WebRotate 360 Product Viewer

USER MANUAL v2.5.3

Rev: July 15th, 2011

Contents

- About the Product 3
- Why WebRotate 360? 3
- Creation process 4
- Viewer components and API 6
- XML configuration 8
- SpotEditor Guide 15
- Using template configuration 18
- Preparation of 360-degree images 19
- FREE version limitations 20

❖ About the Product

WebRotate 360 Product Viewer is a simple and powerful tool for showcasing your products online.

WebRotate 360 Product Viewer does not require any server-side technologies because it consists of just a couple of JavaScript and Flash files running on your web pages. When a page is loaded, the code contained in these files automatically downloads all configured images and resources from your web server and renders them seamlessly according to your custom configuration. You don't have to know JavaScript or Flash to make it work. All you need is to change XML values using any text editor (like Notepad or TextEdit) **or use provided SpotEditor utility to make viewer configuration a breeze**.

Download the latest viewer update on our website at: www.webrotate360.com/360-product-viewer.html

❖ Why WebRotate 360?

- Fine-tuned for E-Commerce
- Works on iPad, iPhone, and other devices and browsers without Adobe Flash
- No server side scripts required - all client based
- Simple and lightweight
- All configuration is done via a separate XML file
- Highly customizable (sizes, features, colors, etc.)
- Extensive hot-spot support with granular control
- Free software utility for quick configuration
- Use with any number of websites / products
- Host on your servers - no extra fees or vendor lock-in
- Dedicated full-time support (PRO)
- Available as FREE and PRO editions
- We are here to help with your customization requests

This technology comes to you from our interactive studio that processed more than 150,000 product images - it's the same viewer we use in our client projects.

❖ Creation process

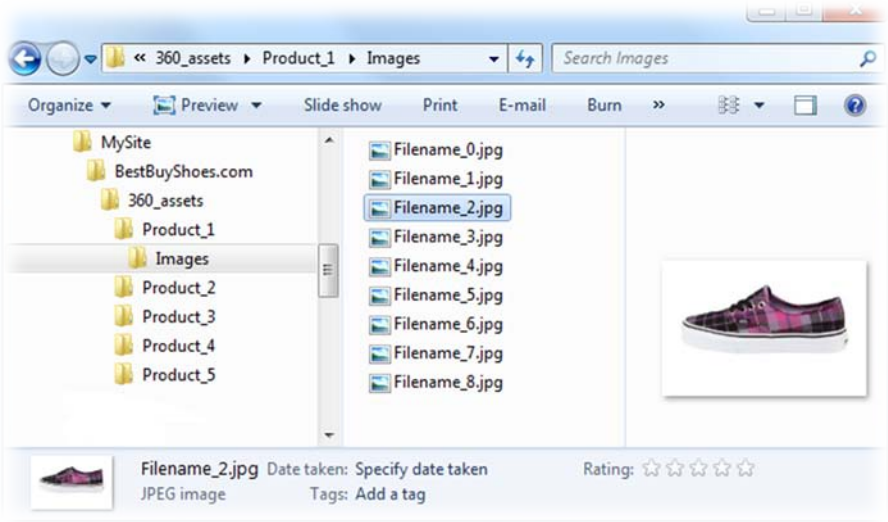
Creation of product views is done in several steps:

- Preparing product images per your requirements (resize, crop, retouch).
- Copying images into a dedicated folder.
- Configuring XML configuration file either manually or using supplied SpotEditor utility.
- Publishing.

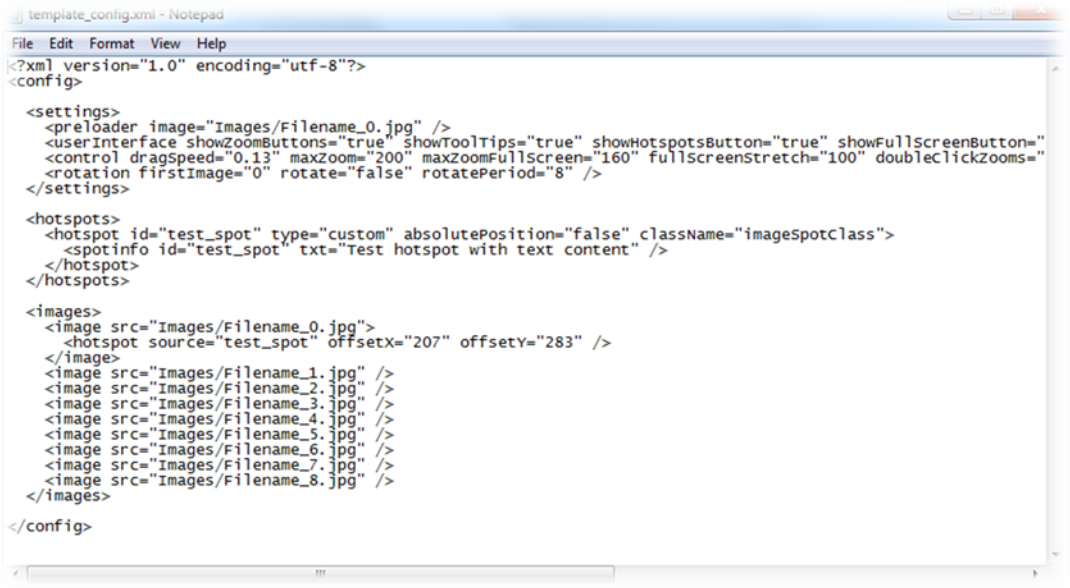
Let's review each of these steps:

WebRotate 360 product viewer uses existing images that you or your photographer will need to capture first. You can use any number of 360-degree images or even just a single image (*in case you need to showcase your product features with just interactive hot-spots and without 360-degree rotation*). Images may need to be re-sized or cropped for better presentation and faster loading and you can use your favorite image editing software (e.g., Adobe Photoshop) to achieve desired results. Read more about recommendations on image numbers and dimensions below (Preparation of 360-degree Images).

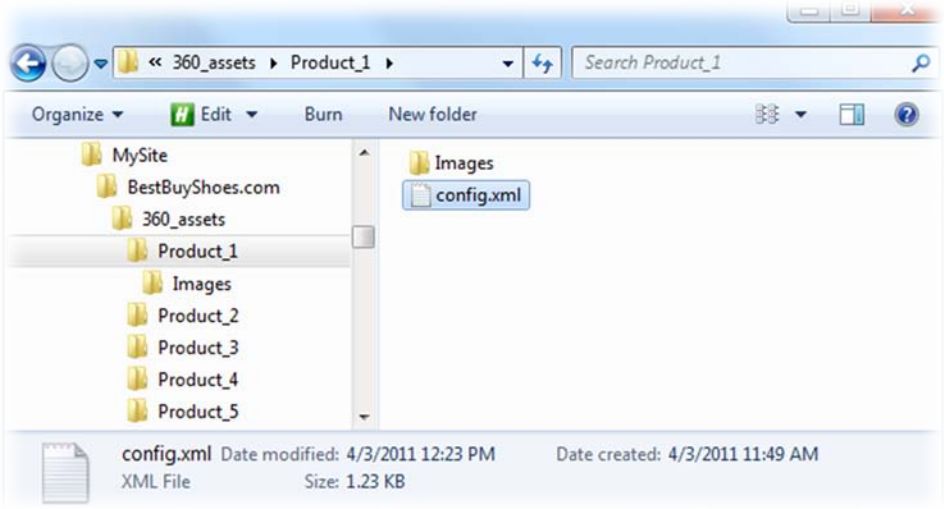
Once images are ready, copy them into a folder on your hard drive that you will use for publishing to your web server. For example you can create a folder for each of your products and copy images under the Images folder under corresponding product folder as follows:



Next step is to create an XML configuration file that will specify the images to be loaded by the viewer, their relative location, hot-spots, user controls, and more. To create XML configuration file for each of your products, you can either use a template XML file supplied with the viewer package (Resources/template_config.xml, as shown on the picture below) or use the SpotEditor utility to streamline the configuration process (read more about the utility below).



Each such configuration file can be stored under corresponding product folder. Note that the SpotEditor utility always creates configuration xml on the same level with the selected image folder as shown on this picture:



The last step is to add a simple piece of JavaScript code to your HTML page or a server side script (e.g., PHP) and reference included CSS and JS files as follows:

```
<head>
<title>WebRotate 360 Viewer Integration Sample</title>

<link type="text/css" rel="stylesheet" href="imagerotator/html/css/basic.css"></link>
<script type="text/javascript" src="imagerotator/html/js/jquery-1.4.2.min.js"></scr
<script type="text/javascript" src="imagerotator/html/js/swfobject.js"></script>
<script type="text/javascript" src="imagerotator/html/js/imagerotator.js"></script>


</head>

<body>


<div id="content" style="width:400px; height:320px; border: 1px solid #cecfd2; ">

    <div id="wr360PlayerId" class="wr360_player">
    </div>

    <script language="javascript" type="text/javascript">

      _imageRotator.settings.jsScriptOnly  = false;
      _imageRotator.settings.swfFileURL    = "imagerotator/imagerotator.swf";
      _imageRotator.settings.configFileURL = "360_assets/SampleShoe/config.xml";

      _imageRotator.runImageRotator("wr360PlayerId");

    </script>

</div>

</body>
```

1 Add CSS and script includes

2 Specify wrapping DIV dimensions that will be used to automatically resize the viewer

3 Add empty DIV with class "wr360_player" and assign its id

4 Copy the following script block somewhere below the DIV and update imageRotator settings accordingly.

NOTE: The new version of SpotEditor utility included with the 2.5.3 viewer package allows creating a complete set of redistributable 360 viewer files with images including sample HTML preview code as shown on the screenshot via Preview button (see more information below in the document).

❖ Viewer components and API

WebRotate 360 product viewer consists of two primary modules - Flash SWF component (imagerotator.swf) and a JavaScript component (imagerotator.js). The two are working together in such way that Flash component is only loaded when visitor's browser has Flash plugin installed (90% of all computers). If Flash is not supported on visitor's computer or a device as in the case with Apple iPad and iPhone, the Flash component is automatically and seamlessly disabled and the JavaScript module is activated instead. You can still configure the viewer to only run in JavaScript mode even when Flash is available.

Note that the JavaScript mode is still limited in functionality and currently doesn't support several features such as zoom, full-screen, and dynamic hot-spots. We are working on adding these features in the future releases.

Following is the list of viewer settings that you can specify on your web page:

_imageRotator.settings.jsScriptOnly

Specify whether Flash component should be disabled even when the Flash plugin is present in visitor's browser. Values: true / false. Default is false.

_imageRotator.settings.swfFileURL

Relative path to imagerotator.swf which by default is located under the imagerotator folder.

_imageRotator.settings.configFileURL

Relative or absolute path to XML configuration file for a given product.

_imageRotator.settings.rootPath

This optional setting can be used to specify alternative location of all viewer assets (images, hot-spots) configured in xml and passed via configFileURL.

_imageRotator.settings.flybyJsMenuFix

This optional setting is only applicable to the Flash component and is false by default. Set it to true if you experience issues where WebRotate's 360 viewer interferes with your website's JavaScript dynamic ("fly-by") menus by overlapping / hiding them.

Initiate viewer on your web page by calling this JavaScript method and passing CSS id of the DIV element with the class name wr360_player:

_imageRotator.runImageRotator ("wr360PlayerId");

If you need to dynamically reload the viewer on the same page with different images without refreshing / reloading the page, you can do this by calling the following JavaScript method and passing a new configuration URL and optionally rootPath if required:

_imageRotator.reload ("360_assets/NewProduct/config.xml");

or

_imageRotator.reload ("NewProduct/config.xml", "http:/CDN/360/NewProduct");

❖ XML configuration

WebRotate 360 product viewer is designed to load product images asynchronously from your web server to ensure smooth user experience and ease of maintenance and support. A simple text-based XML configuration file is used to configure all required images and other resources (e.g. hot-spots) that viewer will need to load when rendered on your web pages.

The same XML configuration file is used to specify a variety of viewer options such as user controls, hot-spots, zooming, rotation parameters, colors, and other features. Please see a sample configuration file located under 360_assets/SampleShoe to review the XML format and default values.

USER INTERFACE <userInterface>

preloader image

Relative path to an image that viewer downloads and shows first. Path is relative to imageRotator.settings.configFileURL (or imageRotator.settings.rootPath) supplied on a web page (see "Viewer components and API"). Other supported values: none / first.

userInterface showZoomButtons

Show / hide zoom button. Values: true / false.

userInterface showHotspotsButton

Show / hide hot-spots on/off switch button. Values: true / false.

userInterface showFullScreenButton

Show / hide full-screen button. Values: true / false.

userInterface showTogglePlayButton

Show / hide play button. Values: true / false.

userInterface showArrows

Show / hide left and right arrow (rotational) buttons. Values: true / false.

userInterface showToolTips

Enable / disable popup hint (tooltip) text on mouse hover over controls. Values: true / false.

userInterface toolbarAlign

Align bottom controls (toolbar controls) all together to the left, right or center of the viewer. Values: left, right, center.

userInterface toolbarForeColor

Color of toolbar controls. Values: html color (example: #999999).

userInterface toolbarHoverColor

Color of toolbar controls on mouse over. Values: html color (example: #999999).

userInterface toolbarAlpha

Transparency of toolbar controls. Values: 0 to 1 (examples: 0.3, 0.67, 0.85).

userInterface toolbarBackColor

Color of toolbar background. Values: html color (example: #999999).

userInterface toolbarBackAlpha

Transparency of toolbar background. Values: 0 to 1 (examples: 0.3, 0.67, 0.85).

userInterface progressLoopColor

Color of progress indicator . Values: html color (example: #999999).

userInterface progressNumColor

Color of progress indicator percentage numbers. Values: html color (example: #999999).

userInterface flashEmbedFonts

If set to false, Flash viewer doesn't use embedded fonts which makes it possible to show hot-post text with different character sets (e.g., Cyrillic). Values: true / false.

CONTROL < control>

control maxZoom

Zoom-in depth in percent. The setting applies when user clicks the Zoom-in button in regular mode (not full-screen). Minimum value is 100, which is equivalent to no zoom (1:1). Values: Numeric, starting with 100. Free Edition max is 200 (i.e., x 2).

control maxZoomFullScreen

Zooming depth in percent in full-screen mode. The setting applies when user clicks the Zoom-in button in full-screen mode only. Minimum value is 100, which is equivalent to no zoom (1:1). Values: Numeric, starting with 100. Free Edition max is 200 (i.e., x2).

control fullScreenStretch

Control whether images have to be stretched automatically when entering full-screen mode. This setting is useful in the situations where images are too small to show as-is in full-screen mode. Minimum value is 100, which is equivalent to no stretch (show as-is, 1:1). Values: Numeric, starting with 100. If setting is not present or equals 0, images are stretched to fill entire full-screen area.

control doubleClickZooms

Enable mouse double-click on images for zoom-in / zoom-out action. Values: true / false.

control disableMouseControl

Disable image dragging / rotation via mouse. Values: true / false.

control dragSpeed

Used in conjunction with rotatePeriod (see below). This setting represents a factor that is multiplied by rotatePeriod to calculate the speed of rotation using mouse dragging or left / right arrows buttons. Values: Numeric, should be between 0 and 1.

ROTATION < rotation>

rotation firstImage

Define image that is shown when all images are fully loaded. In most applications this number will point to the same image as specified under <preloader image>. Values: Numeric, counted from 0 which is the first image specified under <images> (see below).

rotation rotate

Start rotation automatically when all images a fully loaded. Values: true / false / once.

rotation rotatePeriod

The number of seconds for rotation to go one full loop when initiated via the Play button or started via auto play. Speed of rotation when using mouse drag or arrow buttons is different and is calculated by multiplying rotatePeriod by the dragSpeed factor as described above.

rotation bounce

If set to true will force viewer to bounce / stop rotation when it reaches the last configured product image. This setting can be used when a product was not photographed in a full 360 rotation (e.g. 180-dgree rotation). Values: true / false

IMAGES < images>

images

Each image is defined in XML by its <image> element. The only supported image attribute is src. This attribute (src) represents a relative path to image location. Path is relative to imageRotator.settings.configFileURL (or imageRotator.settings.rootPath) defined on a web page (see "Viewer components and API"). Note that the images can be either in JPG or PNG format.

HOTSPOTS < hotspots>

hotspots

Parent node for all hotspots. Each hot-spot is represented via a child <hotspot> element under the parent.

hotspot id

Hot-spot's unique identifier. The id can be any text with no whitespaces (case insensitive).

hotspot type

Reserved for future use. Values: custom

hotspot absolutePosition

Determines whether hot-spot stays in a fixed position as defined by offsetX and offsetY (i.e static hotspot) or moves with the images (dynamic hot-spot). Values: true / false

hotspot className

Determines hot-spot presentation type. Values:

WebRotate360.ImageRotator.StaticHotspotPresenter - static hot-spot presented by its content image or text as specified under <spotinfo> element (see below).

WebRotate360.ImageRotator.LoopCircleActionHotspot - dynamic hot-spot presented as a constantly animated circle with blur effect.

imageSpotClass - dynamic hot-spot presented as a circle with plus that is animated once on load and deactivation (e.g., mouse-out).

hotspot color

Fill color of the hot-spot circle. Values: html color (example: #999999).

hotspot alpha

Transparency of the hot-spot circle. Values: 0 to 1 (examples: 0.3, 0.67, 0.85).

hotspot offsetX

Fixed X coordinate (in pixel) that applies only to hot-spots with absolutePosition set to true. It's relative to the left-top corner of the viewer.

hotspot offsetY

Fixed Y coordinate (in pixel) that applies only to hot-spots with absolutePosition set to true. It's relative to the left-top corner of the viewer.

hotspot disabled

Specifies whether to hide / disable current hot-spot. This is useful when you need to hide a hot-spot without actually deleting its XML configuration. Values: true / false.

hotspot <spotinfo>

Describes hot-spot content based on the content type. Viewer supports the following types of hot-spot content:

TEXT CONTENT - rendered as an area with text using configurable width, font height, color and background as specified in <spotinfo> element:

spotinfo **txt** - text of the hot-spot content.

spotinfo **txtWidth** - fixed width of the text wrapping box.

spotinfo **txtBkColor** - color of the text background.

spotinfo **txtColor** - color of the text (html format; example: #99999).

spotinfo **fntHeight** - font height.

spotinfo **url** - URL of a page to browse on click.

spotinfo **urlTarget** - can be _blank to open in a new window or _self to open in the same browser window / tab.

IMAGE CONTENT - rendered "as-is" using supplied image in src attribute:

spotinfo **src** - relative path to content image (PNG or JPG).

spotinfo **url** - URL of a page to browse on click.

spotinfo **urlTarget** - can be _blank to open in a new window or _self to open in the same browser window / tab.

FLASH CONTENT - rendered "as-is" using supplied Flash component in src attribute:

spotinfo **src** - relative path to the location of Flash component (SWF).

spotinfo **url** - URL of a page to browse on click.

spotinfo **urlTarget** - can be _blank to open in a new window or _self to open in the same browser window / tab.

Hot-spots with either Image or Flash content can also define the attribute of text content. In this case the text is rendered just below a hot-spot image or flash graphic with a small bottom margin and centered horizontally. Each dynamic hot-spot (i.e., absolutePosition = false) is also defined separately under corresponding <image> element. Supported attributes for the hotspot element when specified under <image> are:

hotspot **source** - should be exactly the same as the id of this hot-spot under <hotspots>

hotspot **offsetX** - X coordinate in pixel relative to the left-top corner of an image.

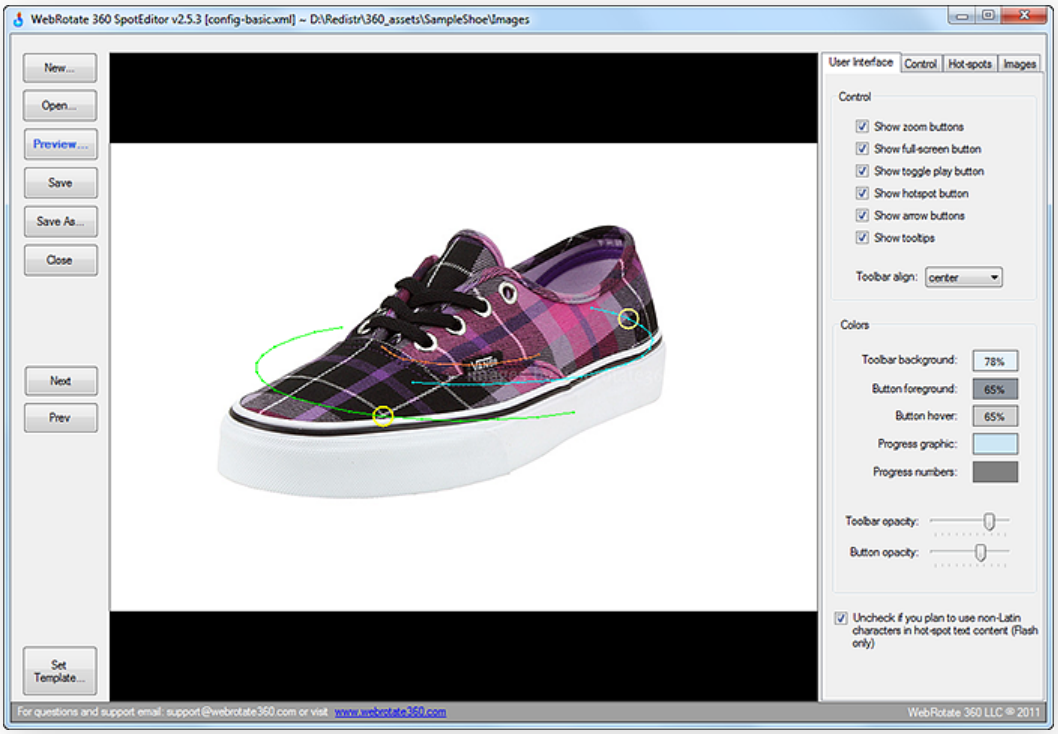
hotspot **offsetY** - Y coordinate in pixel relative to the left-top corner of an image.

NOTE: Configuring images and hot-spots manually via XML can be very time consuming. That's why we also provided a free SpotEditor utility (see Setup.exe under Software) that can dramatically simplify creation of XML configuration with hot-spots. For more information about this utility, please see the following section.

❖ SpotEditor Guide

SpotEditor utility provides a simple and convenient way for creating full-featured 360 product views with hot-spots with just a few mouse clicks. You can create an XML configuration by simply pointing it to a folder with images on your hard-drive and hitting Save, or use the Preview button to generate a complete set of redistributable files with images and configuration that you can test in your local browser or move as-is to you web server.

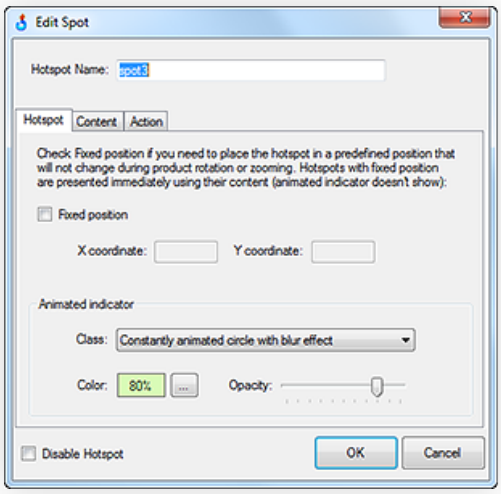
SpotEditor also supports templates that make it possible to use existing XML files as templates for new product views thus saving a lot of time on projects with a large number of views that share the same settings and hot-spots.

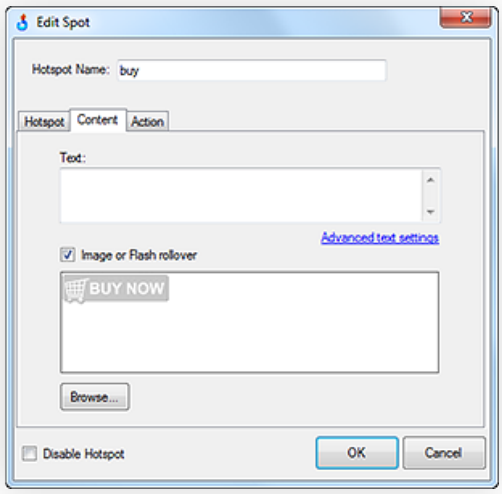


Below is a quick guide on how to install and use the utility:

- Install SpotEditor by executing Setup.exe provided under Software. It may ask to install Microsoft .NET components if not available on your computer already. A shortcut will be placed on your desktop upon successful software installation.
- Start the software by clicking on the new desktop shortcut.
- Select New and use the folder browser dialog to select a folder with product images. Once selected, all images from this folder will be loaded into the application window according to their filename numbering. You will be able to review each image by simply scrolling your mouse wheel, using the Space key on your keyboard or via the Next / Prev buttons on the left of the application window.
- Adjust viewer settings on the User Interface and Control tabs on the right according to your preferences.

- Navigate to the Hot-spots tab and select the New button to create a new hot-spot. Use the Edit Spot dialog to configure hot-spot name, fixed position (if desired), rollover content / action, and other hot-spot presentation settings. Note that hot-spot rollovers (images and Flash swfs) are copied into a new folder called spots that will be created under the same folder with the product images that you originally selected and will be referenced in the configuration file upon saving.





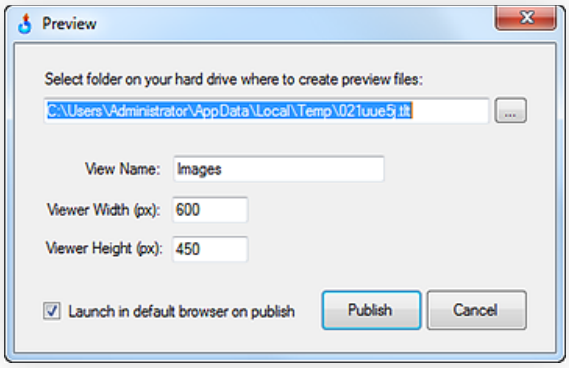
- While on the Hot-spots tab, place a new spot on the currently selected image via left mouse click. You can drag the spot on the image with your mouse, move it via Ctrl + Up /

Down / Left / Right arrow keys. You can also delete it with the Delete key or via right-click context menu. Proceed to the next image either by rolling your mouse wheel, using the Next / Prev buttons on the left, or via the Space key on your keyboard.

- Use the Path Details controls on the Hot-spots tab to set color and opacity of hot-spot trajectories and spot selection circles to make it more visible on your images. If you have many hot-spots, it's sometimes useful to hide other hot-spot trajectories while you are working on the current path. You can do this by selecting the Hide inactive path checkbox under the Path Details.

- To add a custom logo image to the left-top corner of your product viewer, create a new hot-spot, check Fixed position and assign X, Y coordinate (for example 5 and 5). Then select your logo image (and the action URL if required) on the Content and Action tabs accordingly.

- Select Save to create a new XML configuration file. It will be saved on the same level with the image folder that you originally selected. To preview or publish your changes, click Preview button and specify your output folder, desired viewer dimensions and View Name. Then select Publish to generate a complete viewer package and automatically open the viewer in your default browser. You can adjust the settings further and re-publish to your destination location again. Published files can be used for integrating with your website or copying to your web server for testing "as-is".



- Use Open button to open and edit existing XML configuration. You can also use drag and drop to quickly open existing configurations by dragging them with your mouse from Windows Explorer onto the application window. You can also use Save As command to save current XML configuration using a different file name.

NOTE: XML configuration file is always saved on the same level with your image folder. This way we can ensure that the relative location of the images that is saved in the XML file remains constant and will not break as you copy the views from local folders on your computer to the web server.

❖ Using template configuration

SpotEditor installer places a default config.xml under **Program Files\WebRotate 360\SpotEditor**. This configuration file is used by default as a template when configuring new product views. You can modify this default XML configuration file and set your custom template parameters that will be used by default when creating new product views. Another option is to select a custom template (which can be any XML configuration file you have created previously) using the Set Template button on the left side of the application window. All viewer settings including hot-spot information will be carried over to a new product view when creating from a template XML configuration.

NOTE: we're are constantly improving the SpotEditor utility. Please visit our Blog or subscribe to our News Letter on our home page for up to date information on our latest updates and releases.

❖ Preparation of 360-degree images

In our 360 product photography projects we usually produce 20-36 images per product and this is what we recommend to most of our clients. 20 images in particular offer a good balance between smoothness of the animation, download / wait times, and the amount of required photography and post-production efforts.

The next question to consider is viewer dimensions when placed on your webpages and its zoom depth. For example, if your viewer is configured to be 500 x 375px on the product pages, and you want to support x 2 zoom, the images should be at least twice the size of the viewer to present in good quality when zoomed in, or about 1000 x 750px (**NOTE**: **for sharpest product presentation we recommend producing 360 images that are just a bit larger than what they would appear when zoomed in full**). This will result in JPG images being ~80-100 KB per image for an average product image with no background and with a decent JPG quality, which is a ~2 MB download for 20 images. For the average broadband cable speeds at about 3 Mbit/sec (read more), this is ~4-5 sec wait time for a complete view to load. This wait should work well for the majority of e-commerce applications. Note that WebRotate 360 viewer downloads and shows the first pre-loader image immediately so your webpage visitor has something to look at while the rest of the images are downloaded and processed.

Same applies to the full-screen mode where to keep the original image sharpness you will need to produce images that are noticeably larger than your viewing box on the product pages. WebRotate 360 viewer has two settings that control image presentation in full screen mode (see the XML Configuration section above).

When saving images, always use the Save for Web option if available so you can optimize images specifically for Web presentation. Don't over-optimize as even high quality JPG compression (~80%) will result in relatively small files that should work for most 360 product presentations.

Note that due to the zooming algorithm used by the Flash player, it's important to keep image width to height ratio same or similar to the ratio of the viewer. This will ensure that you have sharpest presentation of the images inside the viewer. For example, if your viewer is 480 x 320px, the images can have either the same size or smaller (for best quality, if no zoom/full-screen is required) or stretched proportionally, i.e. 576 x 384, 720 x 480, 960 x 640, etc.

NOTE: **here at WebRotate 360 we have been producing 360 product photography and interactive product presentations for several years. More than 150,000 images have been processed by us to date. We helped clients such as Glock, Kids II, Johnson Outdoors, Meade, Triton Showers, and alike. Hire us for your next 360 product photography project or use our unique skills and expertise to build specialized interactive product presentations just for you!**

❖ FREE version limitations

- Powered By link in the right top corner
- Maximum 4 hot-spots
- Limits zoom depth to x 2
- Limited customer support

Support our work by purchasing unlimited PRO edition at: www.webrotate360.com/360-product-viewer-buy.html

Visit our website for more information about latest software updates. You can also subscribe to our News Letter on our home page, via RSS feed on our Blog , follow us on Twitter or Facebook.

For support and feature requests please e-mail us at support@webrotate360.com